CSCI 251: Concepts of Parallel and Distributed Systems

Alvin Lin

October 4th, 2017

Practice Problem

Illustrate the user of recursive locks using a binary tree search algorithm. The program takes in a large list of numbers. The list is divided across multiple threads. Each thread tries to insert its elements into the tree by using a single lock associated with the tree.

1. Show that the single lock becomes a bottleneck even for a moderate number of threads.
   (This problem is obvious, a single thread inserting an element locks the entire tree and prevents any other thread from writing.)

2. Improve the above program by associating a lock with each node in the tree (as opposed to a single lock with the entire tree). A thread locks a node when it reads or writes it. Examine the performance properties of the implementation.

Practice Problem

Implement a threaded hash table in which a single lock is associated with a block of k hash table entries. Threads attempting to read/write an element in a block must first lock the corresponding block. How does the performance of your algorithm vary as a function of $k(k < n)$, where n is the total number of entries in the hash table.

Practice Problem

The odd-even transposition algorithm sorts N elements in N phases where N is even. The algorithm alternates between two phases called odd and even phases. Each phase requires $\frac{N}{2}$ compare-exchange operations. During the odd phase, elements with odd indices are compared with their right neighbors and they are exchanged if they are out of sequence. Similarly, during the even phase, elements with even indices are compared with their right neighbors, and they are exchanged if they are out of sequence. After N phases of odd-even exchanges, the sequence is sorted.

1. Assume availability of $P = N$ processes. Show a parallel formulation of the problem and compute the parallel time and speedup.
2. Assume availability of $P << N$ processes. Show a parallel formulation of the problem and compute the parallel time and speedup.

Practice Problem

Develop a parallel formulation of the quicksort algorithm. Provide analysis of your algorithm. Show the parallel time and speedup. Discuss the implementation of the algorithm on shared memory and message passing systems.

Reminders

The midterm is on October 11th. Refer to MyCourses for details on Project 1, which is due Friday, October 6th.

Professor Mohan Kumar:
mjkvcs@rit.edu
https://cs.rit.edu/~mjk

Rahul Dashora (TA):
rd5476@mail.rit.edu

You can find all my notes at http://omgimanerd.tech/notes. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech